

Trabalho Teórico Prático

Teoria dos Grafos e Computabilidade

César Luiz Vales Teixeira¹, Hudsson Davih Apolinario de Andrade², Luís Fillipe Magalhães Conrao

¹Pontifícia Universidade Católica de Minas Gerais (PUCMG)
– Belo Horizonte – MG – Brasil

***Resumo.** Este artigo é a documentação do trabalho teórico prático realizado na matéria de Teoria dos Grafos e Computabilidade ministrada pelo professor Gabriel Barbosa da Fonseca. O artigo tem por objetivo explicar e documentar as decisões tomadas ao longo do trabalho.*

1. Linguagem adotada

Todo o trabalho foi desenvolvido majoritariamente na linguagem de programação Java. Ao longo do desenvolvimento do trabalho foi necessário recorrer a elementos da linguagem Python para a geração dos gráficos quantificarmos os resultados obtidos e também para a exportação do dataset utilizado ao longo do desenvolvimento do trabalho por meio da biblioteca sklearn.

2. Biblioteca de Grafos

A primeira parte do trabalho consiste em uma biblioteca para manipulação de grafos. Ela possui uma classe principal, Grafo, que possui métodos para tratar arestas e vértices como definir peso e nome para cada vértice e para cada aresta, adicionar e remover arestas, testar se existe determinada aresta, se determinados vértices ou arestas são adjacentes, se o grafo é completo, se é vazio, entre outros. Ao instanciar um grafo é possível escolher a implementação, sendo as duas opções lista de adjacência e matriz de adjacência. As duas classes ListaAdjacencia e MatrizAdjacencia implementam a interface ImplementacaoGrafo, e elas representam as arestas do grafo usando uma lista e uma matriz, respectivamente. Cada uma utiliza uma classe como aresta: ArestaMatriz e ArestaLista, e as duas classes estendem a classe Aresta. Os vértices são representados por meio da classe Vertice, que assim como as classes de Aresta possuem as implementações necessárias para lidar com nome e peso de vértices ou arestas.

3. Algoritmo de Propagação de Rótulos

O artigo [Dengyong Zhou and Scholkopf 2003], tem por objetivo apresentar um método de aprendizado semi-supervisionado que usa uma função de classificação em relação à estrutura intrínseca revelada por pontos rotulados e não rotulados. O método apresenta um algoritmo simples para obter essa solução e mostra resultados experimentais em vários problemas de classificação.

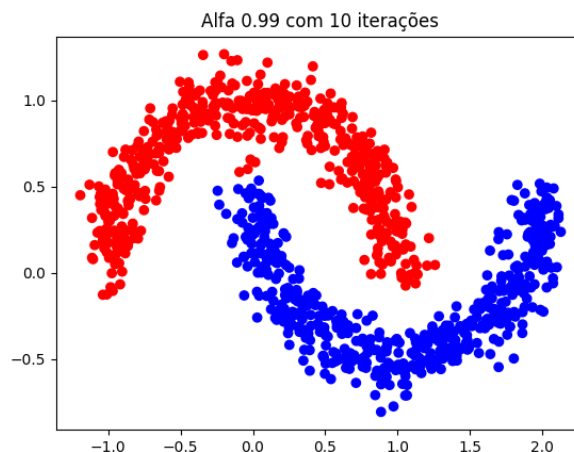
Com base na segunda seção do artigo, implementamos o algoritmo sugerido para obter uma solução para problemas de classificação na linguagem Java como dito anteriormente. O nosso algoritmo inicia-se fazendo a leitura do dataset obtido separando as entradas e os rótulos. Após isso, replicamos o primeiro passo documentado no artigo, onde utilizamos a matriz obtida pelo dataset para calcular a distância euclidiana entre

cada par de pontos de duas matrizes de números reais. Utilizaremos o resultado obtido no método RFB, cujo objetivo é calcular os pesos de uma rede neural baseada em funções de base radial (RBF), funções na qual, dependem apenas da distância entre os pontos de entrada e os centros das funções.

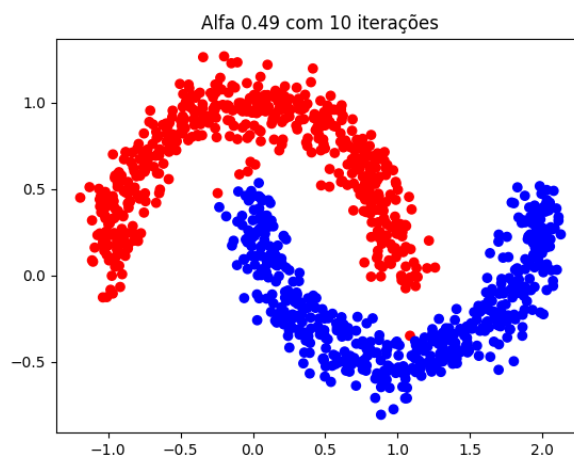
O segundo passo consiste em calcular a matriz de similaridade entre as linhas da matriz W , normalizando-as pelo seu comprimento. Após isso, vamos ao terceiro e último passo, cujo objetivo é calcular a iteração de uma função linear sobre a matriz de entrada, usando um fator de amortecimento alfa, no qual, foi parametrizado em 0.99.

4. Resultados do Algoritmo de Propagação de Rótulos

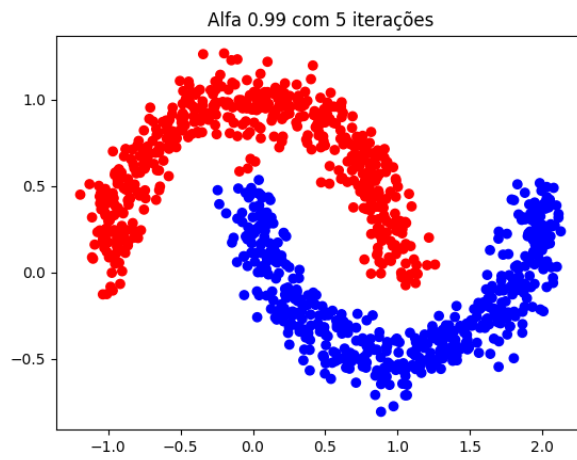
O algoritmo foi testado com diferentes parâmetros para o valor alfa e de iterações. Os resultados obtidos serão mostrados a seguir:



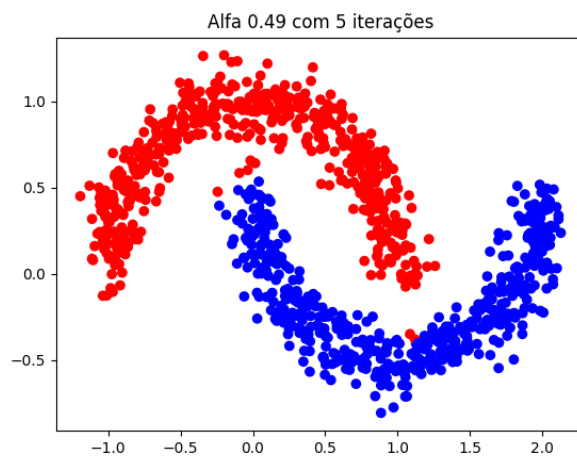
Fonte: Gráfico elaborado pelos autores



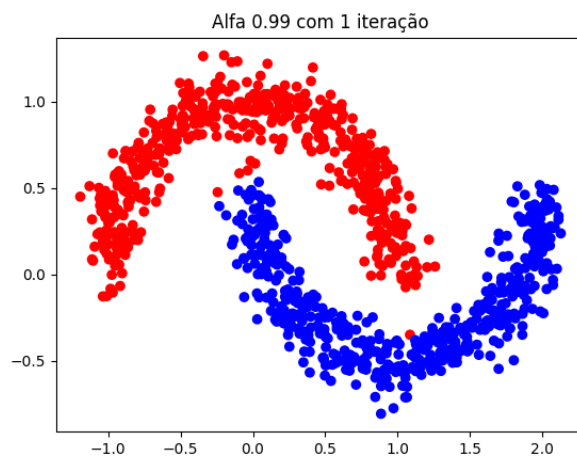
Fonte: Gráfico elaborado pelos autores



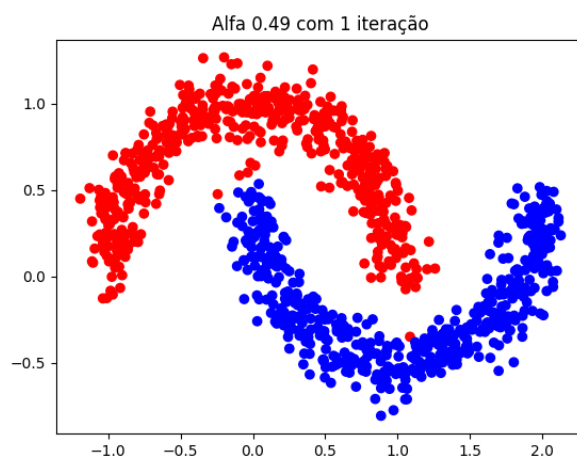
Fonte: Gráfico elaborado pelos autores



Fonte: Gráfico elaborado pelos autores



Fonte: Gráfico elaborado pelos autores



Fonte: Gráfico elaborado pelos autores

Em resumo esse foi os dados obtidos durante a execução do nosso algoritmo:

Table 1. Dados obtidos durante os testes

Iterações	Valor do Alfa	Acurácia
10	0.99	99,6%
10	0.49	99,4%
5	0.99	99,6%
5	0.49	99,3%
1	0.99	97%
1	0.49	97%

Fonte: Tabela elaborada pelos autores

Podemos observar que, mesmo com a variação de iterações e do valor de alfa, o nosso algoritmo se manteve com uma acurácia alta, tendo seu menor resultado encontrado quando simulado com apenas 1 iteração e com um alfa de 0.49.

5. Divisão das tarefas no trabalho

Ao longo do trabalho todos os integrantes do grupo realizaram todas as tarefas em conjunto. Mas em especial César atuou na implementação do algoritmo de propagação de rótulos, Hudsson atuou na biblioteca de grafos e na elaboração deste artigo e Luís Fillipe atuou junto ao César na implementação do algoritmo de propagação e junto ao Hudsson na elaboração deste artigo

References

Dengyong Zhou, Olivier Bousquet, T. N. L. J. W. and Scholkopf, B. (2003). Learning with local and global consistency.